



UNIVERSIDADE ESTADUAL DA PARAÍBA - UEPB

CENTRO DE CIÊNCIAS E TECNOLOGIA – CCT

DEPARTAMENTO DE COMPUTAÇÃO

Dennis Eduardo Santos da Silva - 191080217

Renan Rey Costa Rodrigues - 201080419

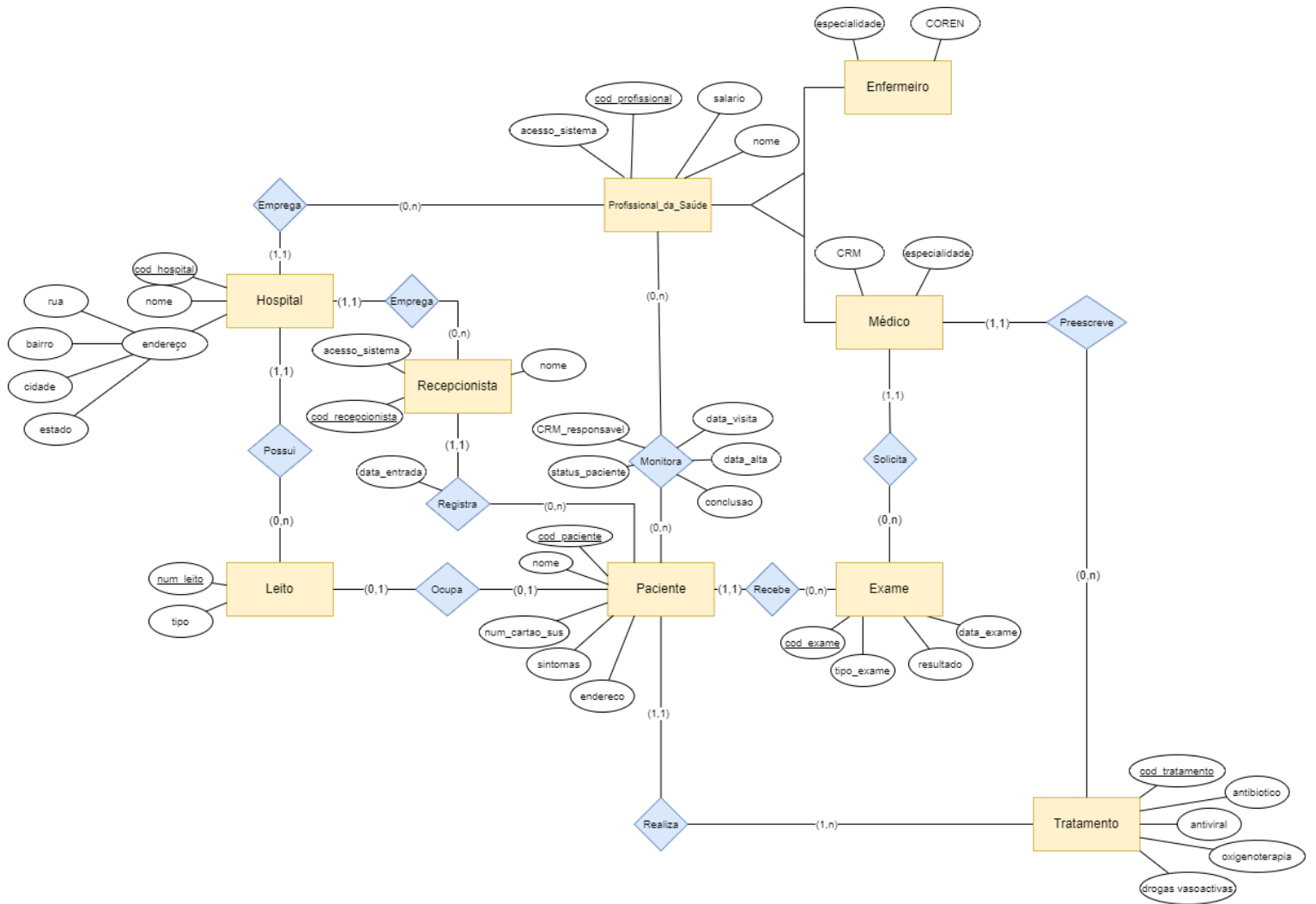
Simeone Aquino De Holanda - 191080446

Projeto Milestones 3 – Observatório da SARS – COV-2

CAMPINA GRANDE - PB

28 de Fevereiro de 2022

Diagrama de entidade e relacionamento:



Projeto Relacional:

Hospital (cod_hospital, nome, descricao, estado, cidade, bairro, rua)

Leito (num_leito, tipo, #cod_hospital*)

Recepcionista (cod_recepcionista, nome, acesso_sistema, #cod_hospital*)

Paciente (cod_paciente, num_cartao_sus*, nome, sintomas, endereco, #cod_recepcionista*, #num_leito, data_entrada, #cod_profissional, #cod_exame)

Profissional_da_saude (cod_profissional, nome, salario, acesso_sistema, #cod_hospital)

Monitora (#num_cartao_sus, #cod_profissional, status_paciente, data_visita, conclusao, CRM_responsavel)

Enfermeiro (COREN*, especialidade, #cod_profissional)

Medico (CRM*, especialidade, #cod_profissional)

Exame (cod_exame, #cod_profissional*, tipo_exame, data_exame, resultado)

Tratamento (cod_tratamento, antibiótico, antiviral, oxigenoterapia, drogas_vasoactivas, #num_cartao_sus, #CRM)

Dicionário de dados:

Entidade: Hospital

Atributo	Classe	Domínio	Tamanho	Obrigatório	Descrição
cod_hospital	determinante	Numérico		Sim	Id identificador do hospital
nome	Simples	Texto	100	Sim	Nome do hospital
descricao	Simples	Texto	máx	Não	Descrição do hospital
estado	Simples	Texto	50	Não	
cidade	Simples	Texto	50	Não	
bairro	Simples	Texto	50	Não	
rua	Simples	Texto	100	Não	

Entidade: Paciente

Atributo	Classe	Domínio	Tamanho	Obrigatório	Descrição
num_cartao_sus	determinante	Numérico		Sim	Cartão único de saúde
cod_paciente	determinante	int		Sim	Id único do paciente
nome	Simples	Texto	100	Não	
sintomas	Simples	Texto	máx	Não	Quadro de sintomas do paciente
endereço	Composto	Texto	máx	Não	Endereço completo
cod_recepcionista	Simples	Numérico		Sim	
data_entrada	Simples	datetime		Não	Data da entrada do paciente no hospital
num_leito	Determinante	int		Não	Id único do leito do paciente
data_visita	Multivalorado	Data		Não	

Entidade: Monitora

Atributo	Classe	Domínio	Tamanho	Obrigatório	Descrição
Status_paciente	Simples	Texto	100		Descrição do status do paciente
data_visita	Simples	Data			Data e hora da visita do profissional de saúde para checar o status do paciente
data_alta	Simples	Data		Não	Data e hora do momento da alta do paciente
conclusao	Simples	Texto	Máx	Não	Descrição do status final do paciente
CRM_responsavel	Simple	int		Sim	CRM do médico responsável naquela situação

Entidade: Profissional_da_saude

Atributo	Classe	Domínio	Tamanho	Descrição
cod_profissional	determinante	Numérico		
nome	Simples	Texto	50	
salário	Multivalorado	Numérico		Formato 00,00R\$
acesso_sistema	Simples	booleano		Booleano indicando se o profissional tem acesso ao Sistema

Entidade: Enfermeiro

Atributo	Classe	Domínio	Tamanho	Descrição
COREN	determinante	Numérico		Sigla do conselho de enfermagem
especialidade	Simples	Texto	200	

Entidade: Médico

Atributo	Classe	Domínio	Tamanho	Descrição
CRM	determinante	Numérico		Sigla do conselho de medicina
especialidade	Simples	Texto	200	

Entidade: Examina

Atributo	Classe	Domínio	Tamanho	Descrição
cod_exame	determinante	Numérico		
tipo_exame	Simples	Texto	50	
data_exame	Multivalorado	Data		Formato dd/mm/aaaa
resultado	Simples	Texto	200	

Entidade: Tratamento

Atributo	Classe	Domínio	Tamanho	Descrição
cod_tratamento	determinante	Numérico		
antibiótico	Simples	Texto	50	
antiviral	Simples	Texto	50	
oxigenoterapia	Simples	Texto	50	
drogas	Simples	Texto	50	
vasoactivas	Simples	Texto	50	

Query criação e alimentação tables:

```
DROP DATABASE IF EXISTS observatorioSARS;
CREATE DATABASE observatorioSARS;

CREATE TABLE hospital (
  cod_hospital SERIAL PRIMARY KEY,
  nome varchar(100) NOT NULL,
  descricao text,
  estado varchar(50),
  cidade varchar(50),
  bairro varchar(50),
  rua varchar(100)
);

CREATE TABLE leito (
  num_leito SERIAL PRIMARY KEY,
  tipo varchar(50),
  cod_hospital int NOT NULL
);

CREATE TABLE recepcionista (
  cod_recepcionista int PRIMARY KEY,
  nome varchar(100),
  acesso_sistema varchar(50),
  cod_hospital int NOT NULL
);

CREATE TABLE profissional_da_saude (
  cod_profissional PRIMARY KEY,
  nome varchar(100),
  salario decimal,
  acesso_sistema varchar(50),
  cod_hospital int NOT NULL
);

CREATE TABLE enfermeiro (
  cod_profissional PRIMARY KEY,
  COREN int NOT NULL,
  especialidade text
);

CREATE TABLE medico (
  cod_profissional int PRIMARY KEY,
  CRM int NOT NULL,
  especialidade text
);

CREATE TABLE paciente (
  cod_paciente int PRIMARY KEY,
  num_cartao_sus int NOT NULL,
  nome varchar(100) NOT NULL,
  sintomas text,
  endereco text,
  data_entrada timestamp DEFAULT (now()),
  num_leito int,
  cod_recepcionista int NOT NULL
);

CREATE TABLE monitora (
  cod_paciente int NOT NULL,
  cod_profissional int NOT NULL,
  status_paciente text,
  data_visita timestamp DEFAULT (now()),
  conclusao text,
  CRM_responsavel int NOT NULL,
  data_alta timestamp
);
```

```

CREATE TABLE exame (
    cod_exame int PRIMARY KEY,
    tipo_exame varchar(100),
    resultado text,
    data_exame timestamp DEFAULT (now()),
    cod_paciente int NOT NULL,
    cod_profissional int NOT NULL
);

CREATE TABLE tratamento (
    cod_tratamento int PRIMARY KEY,
    antibiotico varchar(50),
    antiviral varchar(50),
    oxigenoterapia varchar(50),
    drogas_vasoativas varchar(50),
    cod_paciente int NOT NULL,
    cod_profissional int NOT NULL
);

-- Criando FK
ALTER TABLE leito ADD FOREIGN KEY (cod_hospital) REFERENCES hospital (cod_hospital) ON DELETE CASCADE;
ALTER TABLE recepcionista ADD FOREIGN KEY (cod_hospital) REFERENCES hospital (cod_hospital) ON DELETE CASCADE;
ALTER TABLE profissional_da_saude ADD FOREIGN KEY (cod_hospital) REFERENCES hospital (cod_hospital) ON DELETE CASCADE;
ALTER TABLE medico ADD FOREIGN KEY (cod_profissional) REFERENCES profissional_da_saude (cod_profissional) ON DELETE CASCADE;
ALTER TABLE enfermeiro ADD FOREIGN KEY (cod_profissional) REFERENCES profissional_da_saude (cod_profissional) ON DELETE CASCADE;
ALTER TABLE paciente ADD FOREIGN KEY (num_leito) REFERENCES leito (num_leito) ON DELETE CASCADE;
ALTER TABLE paciente ADD FOREIGN KEY (cod_recepcionista) REFERENCES recepcionista (cod_recepcionista) ON DELETE CASCADE;
ALTER TABLE monitora ADD FOREIGN KEY (cod_paciente) REFERENCES paciente (cod_paciente) ON DELETE CASCADE;
ALTER TABLE monitora ADD FOREIGN KEY (cod_profissional) REFERENCES profissional_da_saude (cod_profissional) ON DELETE CASCADE;
ALTER TABLE exame ADD FOREIGN KEY (cod_paciente) REFERENCES paciente (cod_paciente) ON DELETE CASCADE;
ALTER TABLE exame ADD FOREIGN KEY (cod_profissional) REFERENCES medico (cod_profissional) ON DELETE CASCADE;
ALTER TABLE tratamento ADD FOREIGN KEY (cod_paciente) REFERENCES paciente (cod_paciente) ON DELETE CASCADE;
ALTER TABLE tratamento ADD FOREIGN KEY (cod_profissional) REFERENCES profissional_da_saude (cod_profissional) ON DELETE CASCADE;

-- SEMEANDO TABLES
INSERT INTO hospital (nome, descricao, estado, cidade, bairro, rua)
VALUES ('Hospital da Crianca', 'Hospital especializado no tratamento de crianças', 'Paraiba', 'Campina Grande', 'Centro', 'Rua da Alvorada'),
('UPA 1', 'Hospital emergencial de rápido atendimento', 'Paraiba', 'Campina Grande', 'Alto Branco', 'Rua Manoel Tavares'),
('Trauma', 'Hospital de urgencia e emergencia', 'Paraiba', 'Campina Grande', 'Liberdade', 'Avenida 123')

INSERT INTO leito (tipo, cod_hospital)
VALUES ('Enfermaria', 1),
('Enfermaria', 1),
('Enfermaria', 1),
('Enfermaria', 1),
('Enfermaria', 1),
('UTI', 1),
('UTI', 1),
('Enfermaria', 2),
('Enfermaria', 2),
('Enfermaria', 2),
('Enfermaria', 2),
('UTI', 2),
('UTI', 2),
('UTI', 2),
('Enfermaria', 3),
('Enfermaria', 3),

```

```

('UTI',3),
('UTI',3),
('UTI',3),
('UTI',3),
('UTI',3),
('UTI',3)

INSERT INTO recepcionista (cod_recepcionista, nome, acesso_sistema, cod_hospital)
VALUES (1020, 'Simeone S.', 'admin', 1),
(2040, 'Dennie', 'admin', 1),
(3040, 'Renan', 'admin', 2),
(4060, 'Fabio', 'admin', 3)

INSERT INTO profissional_da_saude (cod_profissional, nome, salario, acesso_sistema, cod_hospital)
VALUES (1, 'Pedro', 2000, 'simples',1),
(2, 'Maria', 2500, 'simple',2),
(3, 'Felipe', 2000, 'simples',2),
(4, 'Ana', 3000, 'simples',3),
(5, 'Rosa', 2000, 'simples',3),
(6, 'Mario', 1800, 'simples',1)
(7, 'Dr. Carlos', 10000, 'intermediario',1),
(8, 'Dra. Fernanda', 8000, 'intermediario',2),
(9, 'Dr. Damiao', 10000, 'intermediario',3)

INSERT INTO enfermeiro (cod_profissional, COREN, especialidade)
VALUES (1, 11111, 'enfermagem da criança'),
(2, 22222, 'enfermagem emergencial'),
(3, 33333, 'enfermagem emergencial'),
(4, 44444, 'enfermagem emergencial'),
(5, 55555, 'enfermagem emergencial'),
(6, 66666, 'enfermagem preventiva')

INSERT INTO medico (cod_profissional, CRM, especialidade)
VALUES (6, 66666, 'Medico da criança'),
(7, 77777, 'Medico geral'),
(8, 88888, 'Medico de emergencia')

INSERT INTO paciente (cod_paciente, num_cartao_sus, nome, sintomas, endereco, data_entrada,
num_leito,cod_recepcionista)
VALUES (1, 1111, 'Fabio', 'Febre e dor de cabeça', 'Rua 123', DEFAULT, NULL,1020),
(2, 2222, 'Pedro', 'covid', 'Rua 456', DEFAULT, 1,3040),
(3, 3333, 'Ricardo', 'covid', 'Rua 77', DEFAULT, 3,3040),
(4, 4444, 'Robson', 'dedo roxo e torcao', 'Rua 888', DEFAULT, NULL,2040),
(5, 5555, 'Renata', 'covid', 'Rua 123', DEFAULT, NULL,4060),
(6, 6666, 'Saulo', 'Febre e dor de cabeça', 'Av 123', DEFAULT, 4,4060)

INSERT INTO monitora (cod_paciente, cod_profissional, status_paciente, data_visita, conclusao,
CRM_responsavel, data_alta)
VALUES (1, 1, 'Sintomas leves', DEFAULT, 'liberado', 66666, '2022-03-10'),
(2, 2, 'Sintomas de covid', DEFAULT, 'paciente internado devido sintomas de covid', 77777, '2022-03-10'),
(3, 6, 'Sintomas de covid', DEFAULT, 'paciente internado devido sintomas de covid', 77777, '2022-03-11')

INSERT INTO exame (cod_exame, tipo_exame, resultado, data_exame, cod_paciente, cod_profissional)
VALUES (1,'covid','positivo', DEFAULT, 3,6)

INSERT INTO tratamento (cod_tratamento, antibiotico, antiviral, oxigenoterapia, drogas_vasoactivas,
cod_paciente, cod_profissional)
VALUES (1, NULL, 'Coquetel Covid', 'Sim', NULL, 3, 6)

-- -- create procedure
-- CREATE FUNCTION trigger_set_timestamp()
-- RETURNS TRIGGER AS $$
-- BEGIN
--     NEW.updated_at = NOW();
--     RETURN NEW;
-- END;
-- $$ LANGUAGE plpgsql;

```



```
-- -- auto updated_at paciente
-- CREATE TRIGGER set_timestamp
-- BEFORE UPDATE ON data_entrada
-- FOR EACH ROW
-- EXECUTE PROCEDURE trigger_set_timestamp();

-- to run DELETE tables
-- DELETE FROM profissional_da_saude;
-- DELETE FROM hospital;
-- DELETE FROM recepcionista;
-- DELETE FROM leito;
-- DELETE FROM paciente;
-- DELETE FROM medico;
-- DELETE FROM enfermeiro;
-- DELETE FROM exame;
-- DELETE FROM tratamento;

-- restart sequence auto-increment from tables ids
-- ALTER SEQUENCE hospital_cod_hospital_seq RESTART WITH 1;
```